Sequential and deterministic programming languages enjoy a comprehensive landscape of semantics, operational and denotational, static or dynamic. Over the last decades, a number of connections between those have been developed. In contrast, the execution of quantum programming languages is rather poorly understood. Of course it can be formulated operationally, see e.g. [6]. However, the denotational picture is not as complete. Static semantics have been developed: Selinger and Valiron proved [9] full abstraction for the linear quantum $\lambda$-calculus using an interpretation in the category of Hilbert spaces and completely positive maps, and later, Pagani, Selinger and Valiron extended this model to an adequate model of the full quantum $\lambda$-calculus [6]. On the dynamic side though, no satisfactory denotational account has been given. Dal Lago et al have developed a dynamic account of the execution of the quantum programming language via the geometry of interaction [5] – their methodology is, however, not compositional. Earlier, Delbecque and Panangaden have given a game semantics for a quantum programming language, with however a rather drastic restriction on the allowed entangled states.

In this work, we present a new framework for the game semantics of quantum programming languages. To do this we draw inspiration on two lines of work. On the one hand, our account of the dynamics of computation comes from the *concurrent games on event structures* of Winskel et al [7, 2, 3], and their extension with probabilities [10, 1] (which it extends conservatively). On the other hand, our formulation of quantum states is inspired by the work of Pagani, Selinger and Valiron [9, 6, 8]. Our basic framework yields a compact closed category of games carrying Hilbert spaces, and strategies whose states carry operators acting on these spaces. We then apply this compact closed category to the semantics of the linear quantum $\lambda$-calculus. In contrast to Delbecque and Panangaden’s work [4], we are able to faithfully represent any entangled state. This is therefore the first denotational dynamic account of quantum programming languages.

References


